RSRE
MEMORANDUM No. 3900

ROYAL SIGNALS & RADAR ESTABLISHMENT

LOW LEVEL SEGMENTATION OF NOISY IMAGERY

Author: R G White

RSRE MEMORANDUM No. 3900

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN.
WORCS.

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3900

TITLE: LOW LEVEL SEGMENTATION OF NOISY IMAGERY

AUTHOR: R G WHITE

DATE: February 1986

SUMMARY

A possible approach to image segmentation is first to perform a low-level segmentation. This then allows an original image to be described in terms of a set of simple regions or primitives. Objects in the image may be subsequently recognised by matching these primitives to patterns of primitives in a data-base. It is found that current techniques for low-level image segmentation fail when applied to high noise images. An algorithm is presented which overcomes the problems associated with high noise and succeeds in generating low-level segementations of noisy imagery. The algorithm is shown also to work on low noise data.

This memorandum is for advance information. It is not necessarily to be regarded as a final or official statement by Procurement Executive, Ministry of Defence

S80/59

MEMORANDUM NO 3900

LOW LEVEL SEGMENTATION OF NOISY IMAGERY

R G White

INTRODUCTION

The ultimate aim of an image segmentation process is to divide an original image into a set of labelled regions. Each of the regions should satisfy some uniformity criteria. This uniformity may take the form of a low-level description, such as the constancy of an underlying image intensity, or a high level description. A high level description may, for example, divide an agricultural scene into woods and fields or a town into buildings and roads. For most image understanding purposes the high-level segmentation will be required as the final output. Such a description could be generated by matching objects in the image with objects in a data-base of knowledge.

The severity of the segmentation problem depends upon the amount of information available. If, for example, scaling and orientation information exists the problem is easily solved by template matching techniques. Unfortunately for the majority of cases much less information is available. Combinational considerations then effectively rule out the use of such simple procedures except for the smallest of images.

In order to generate a high level description of an image these combinational problems must be removed before objects may be matched with the data base. This requires some sort of data reduction. Once a significant data reduction has been achieved then it should become possible to use recognition techniques which are more advanced than simple template matching.

Such techniques may, for example, use a syntactic description of the scene with patterns being subsequently recognised by string matching [ref 1]. Any data reduction method which is to be used in an approach such as this must preserve a very large proportion of the information contained in the original image whilst removing the uncertainties associated with noise processes. Initially then a low-level image segmentation is required to reduce the data content of the image without reducing its information content significantly.

Data reduction should, as mentioned above, effectively merely remove the noise in an image. There are basically two types of noise to be considered, additive noise and multiplicative noise. Additive noise is the type most often encountered. Incoherent imaging systems, such as optical or infra-red devices, exhibit such noise behaviour. However a large class of imaging systems exist which employ coherent illumination. Lasers and radar are both examples of coherent illumination. Images formed using such illumination are characterised by multiplicative noise or speckle [ref 2]. This is a result of the coherent interference of returns from many individual scatterers in the object being viewed. The probability density function for the detected power received from a scene with a uniform background scattering cross-section is

$$f_z(z) = \frac{1}{\mu_z} \exp(-z/\mu_z) \qquad (1)$$

z = detected power

μ_z = expected value of the detected power.

Taking the signal to noise ratio to be defined as $\mu_z{}^2/\sigma_z{}^2$, where $\sigma_z{}^2 = \overline{(z-\mu_z)^2}$ leads to a signal to noise value of 1. Hence a coherent imaging system may be considered as an example of an extremely noisy system. The segmentation of coherent images will mainly be considered in this paper although it is shown that the segmentation procedure described will also work on incoherent imagery.

2. REVIEW OF SEGMENTATION ALGORITHMS

A great deal of research has been carried out into image segmentation techniques. However almost all of this has been applied to incoherent imagery. It is not a trivial extension to move from incoherent data to coherent data and the techniques developed on incoherent imagery usually fail when applied to coherent imagery. A review of existing segmentation methods, given below, demonstrates this fact.

Segmentation techniques may be grouped into three main categories: region fitting, region growing and edge detection. In addition to these three main groupings of segmentation methods various smoothing operations have been developed. These attempt to reduce or remove image noise whilst maintaining the full image resolution.

Region Fitting Methods

Segmentation by region fitting is accomplished by attempting to fit a given primitive template, or family of templates, to each portion of the input image.

One of the simplest implementations of this approach is the split and merge technique [ref 3]. As the name suggests segmentation is achieved by either joining adjacent regions together if they are similar or splitting a region if it is found to be inhomogeneous. Specifically the procedure works as follows. The algorithm begins by splitting the initial square image into a series of square subimages. If any set of 4 adjacent subimages are determined to be sufficiently similar they are merged to form one larger square. Alternatively if any one subimage is determined to be inhomogeneous it is split itself into 4 squares. The algorithm continues in an interative manner until no more splitting or merging occurs.

Although the technique may be applied to coherent data there are several objections to the algorithm. Firstly because the method operates by producing square divisions of the original image then the final image tends also to be compared of regions having a square shape. *Secondly the regions* produced are start point dependent and homogeneous regions may be segmented depending on their spatial position with respect to the square search lattice. Finally the approach tends to lose small regions within otherwise large uniform areas.

In an attempt to remove some of the problems associated with the split and merge technique a facet model approach has been investigated [ref 4]. The facet model assumes that the image domain may be represented by a set of facets {F(1) ..., F(K)}. For each pixel in the facet F(k) the ideal gray tone is taken to be a polynomial function of the pixels co-ordinates. The facet assignments are made by fitting a polynomial function to the pixel values within a square search window. The window is moved in single pixel steps over the full image. Therefore if the window has a size M each image pixel is included in M^2 test windows and a least squares fitting procedure is used to determine the optimum facet parameters from the M^2 choices. Pixels with similar facet parameters are then grouped.

The algorithm performs poorly on areas in an image with length scales less than the window size M. Such small regions are generally broken into single pixels and as such remain effectively unclassified.

To apply the technique to coherent images would require an increase in the window size in order to obtain a reasonable estimate of the polynomial coefficients in the high multiplicative noise environment. Increase in window size would clearly lead to a compounding of the problem outlined above. Consequently the technique is unlikely to be of use in coherent image segmentation.

Grimson and Pavlidis [ref 5] have recognised the failings as indicated above of such fitting procedures. In an attempt to overcome the problem they suggest measuring the residual between the fitted facet and the true data. The form of this residual may be used to indicate when a discontinuity has been encountered therefore removing the small unclassified regions found above. However as Grimson and Pavlidis point out such a method will only work if the discontinuity step is much larger than the image noise. This restriction indicates that the technique would only work on coherent data in certain situations when very large intensity ratios are encountered.

Region Growing Techniques

The major problem with region fitting techniques, as outlined above is the requirement that the data be fitted to a fixed window. The window must not be large otherwise the results become confused when small length scales are encountered. On the other hand the window must be large enough to enable true discontinuities to be detected in the presence of noise. It is these two conflicting requirements which cause the methods to fail on all but low noise incoherent data. Clearly what is needed is a variable window fitting procedure. Such methods are classed here as region growing techniques in that any region is allowed to grow to any desired window shape and size.

In order to achieve this effect Derin et al [ref 6] have applied the Bayes smoothing algorithm to images modelled by Markov random fields. Briefly the method works as follows. Initially a random image is generated. Changes or relaxations are then made in this random image. A new image is therefore generated which is, hopefully, a closer match to the original as determined by some convenient measure of fit. Changes in the image which cause a poorer fit to the data are also allowed. Such changes enable local minima, encountered in the measure of fit, to be overcome. The process is iterated. In order that the final image is not merely a replica of the first constraints are built into the relaxation process to force the result to conform to some initial model. Such a model might be simply that neighbouring pixels should look similar. The particular problem faced by Derin et al was concerned with additive noise but the process is clearly extendable to multiplicative noise.

The approach is extremely powerful and produces good results. However combinational considerations make the full process computationally too expensive. In order to overcome this problem Derin et al [ref 6] have simplified the image procedure. Three main approximations are used. Firstly pixels are assumed only to interact (influence) directly their nearest neighbours. Secondly the image is processed in strips 3 rows wide thus reducing the possible image combinations. Finally the technique is applied only to binary images, again further reducing the possible image combinations.

Using these approximations the algorithm performs well in segmenting high noise test data as well as real coherent data. Unfortunately these short-cuts may be too restrictive to be of general use. However other methods of off-setting the computational load do exist.

The number of calculations required to perform a full segmentation is dependent on the initial guess at the result. To be fully general this initial estimate is taken to be random as indicated above. Computation time may therefore be saved if the process were seeded with the results of a previous segmentation. Such a combination may provide a very powerful segmentation tool, however the problem of generating the initial segmentation needs to be solved.

A different region growing algorithm has been suggested by Oddy and Rye [ref 7] for use on coherent data. This algorithm operates in three stages. Initially a smoothing filter is applied to the image. The filter operates over a 3 x 3 window and the image is smoothed if the mean absolute intensity difference over the window is less than some user supplied threshold. This filter is applied iteratively typically up to 5 times with different thresholds being supplied by the user at each iteration. The second stage following the smoothing is a bonding process where pixels are bonded if their values lie within a user supplied threshold. Finally boundary tracing and filling is performed.

Clearly the major objection to this approach is the need for interactive parameters. It has not proved possible to find fixed optimum parameters which may be used on different images. This problem stems from the way in which the segmentation proceeds with only 8 pixels at most ever being considered. This results in a high sensitivity to noise. The method is therefore of little use.

Edge Detection Methods

The final class of segmentation techniques is that of edge detection. Edge detection techniques have been widely used to segment incoherent images. Examples of edge operator's used in such work are the Robert's, Prewitt and Sobel operators. These efectively calculate the gray level difference over a 2 x 2, 3 x 3 or 5 x 5 mask. All of them work extremely poorly on coherent data [ref 2]. The failure of all these methods is due to the use of only small operator or mask sizes.

Frost et al [ref 2] have tried to overcome this problem by using a larger window (9 x 9). As Frost et al point out the use of a large window can easily introduce edge orientation problems and they suggest a method of avoiding such problems as follows. For each position of the filter a test is made to decide whether it is more likely the filter is lying in an homogeneous region or is overlying an edge between two regions. To do this the pixels within the window are tested to see if they may be fitted best by a single distribution or by two distributions.

The procedure works well with coherent images which have been incoherently averaged over at least 7 independant realisations of the scene (7-look).

For 4-look averaging the results are described as marginal whilst lower degrees of averaging produce poor results. A major drawback to the technique (as pointed out by Frost et al) is the size of the window. Due to the relative complexity of the decisions made in this technique a reasonably large window is required. It is therefore doubtful whether the algorithm will work with imagery containing variable length scales and variable region constrast ratios.

An alternative approach to the edge-detection technique has been proposed by Don and Fu [ref 8]. Don and Fu, like Frost et al, have recognised the inadequacies of simple edge detection operators. The specific problem solved by Don and Fu was that of finding a sea coast boundary. The approach adopted being to initially divide the area into 16 x 16 subimages. Each image is then classified as either sea or land according to a texture measure. The Sobel edge operator was then applied in the vicinity of this roughly detected boundary. Possible true positions of the sea-coast boundary are then followed with a maximum of three candidates being held at any one time. Fair results are obtained.

The specific example used above may of course be generalised. However it may not always be desirable, or even possible to determine the classification criteria used to find the approximate initial boundary. Without this first step the problem would become computationally expensive. Consequently it is thought that such an anproach will not be generally useful.

It is obvious, if edge detection is to succeed, that an approach be adopted which uses a variety of window sizes. A segmentation scheme employing this idea has been suggested by Rosenfeld and Thurston [ref 9, ref 10]. Their algorithm works as follows. Initially the image is convolved with a family of edge operators which simply calculate the intensity difference between two adjacent neighbourhoods. Neighbourhood sizes (d) are 1, 2, 4, 8 .... 2^{k-1}. Each pixel in the image therefore has k edge values. The problem now arises as to which edge value to adopt. The edge value is chosen as the value belonging to a neighbourhood size d for which the next smallest neighbourhood does not produce a significant improvement in the intensity difference. Significant here is taken as > 4/3. The procedure is outlined for a one dimensional case in figure 1 where the size of the region depicted is in fact the worst size as far as this method is concerned.

In the absence of noise a window size of 8 would be selected here. However only a slight noise perturbation would be required to cause a window size of 16 to be chosen. It may be seen that in a high noise environment the outputs of the various window sizes would be too unreliable to allow the method to work successfully. Even without this added problem associated with high noise values the method is only partially successful. Rosenfeld and Thurston indicate that the method will only segment small isolated regions or large uniform regions. Such a shortcoming is only to be expected because the output values for neighbourhoods covering more than one region will be highly confused. The technique is therefore of no use for the segmentation of real coherent imagery.

So far the edge-detection techniques considered have generally used square or rectangular windows with sharp edges. An alternative approach has been used by Marr and Hildreth [ref 11], amongst others. The image (I), in this technique, is convolved with a two-dimensional Gaussian function

$$G(r) = (1/2\pi\sigma^2)\exp(-r^2/2\sigma^2) \qquad (2)$$

Potential edges are initially found by applying the Laplacian operator $\nabla^2 = \frac{1}{r}\frac{d}{dr}\left(r\frac{d}{dr}\right)$ to the convolved image and locating the zeros.

Several different sizes of Gaussian filter are used (different σ's) and the final edge output is taken to be those points where the zeros from the separate convolved images overlap. This technique might be expected not to work for the following reasons. Firstly large value of image noise are likely to generate many zeros in $\nabla^2(G*I)$ which do not correspond to true edges. Secondly when the Gaussian filter is of a size comparable to that of regions in the image the various zeros will be displaced away from the true edges.

The latter point is accepted by Marr and Hildreth as a shortcoming. However the noise encountered by Marr and Hildreth did not necessitate the need for wide Gaussian filters. Consequently regions of interest in their images nearly always has length scales larger than the filters. In a high noise environment this will not be the case and the problem will be severe.

The problem of spurious zeros has been shown by Giess [ref 12] also to be severe in coherent imagery. Consequently the technique is of no use for the low level segmentation of coherent data. This has been demonstrated by Giess [ref 12].

Smoothing Algorithms used to reduce noise effects

In all the cases considered above it is the nature of the multiplicative noise in coherent imagery which makes segmentation difficult. As mentioned earlier several attempts have been made to reduce the effects of the noise by various smoothing algorithms. One such smoothing procedure was used by Oddy et al [ref 7] in their region growing algorithm described above. This approach produced poor results. Indeed it has been shown that smoothing filters in general offer little or no aid to segmentation procedures [ref 13 ref 14]. Such a result might be anticipated. If accurate decisions can be made in the smoothing process, using a given window size, as to whether an edge exists or not then it should also be possible, in principle, to perform a reasonable, if not quite perfect, segmentation directly using the same size of window.

In summary, therefore, no satisfactory technique exists for the initial low-level segmentation of coherent, or high-noise incoherent, imagery. This failure is a direct result of the multiplicative noise in coherent imagery. The high noise figures generated by this process mean that large areas of pixels must be averaged to obtain accurate estimates of the underlying distribution mean. The averaging requirement means the methods suffer from either loss of resolution, loss of performance or excessive computational expense.

The segmentaticn procedure outlined below overcomes these shortcomings to a great extent. In addition to this it is found to be generally applicable to not only coherent data but to incoherent data as well.

3. THE SEGMENTATION ALGORITHM

The segmentation method outlined below is essentially an edge detection process. Consequently the technique has a global model of the world implicitly built in. This model assumes that scenes are composed of different connected regions. Each region it is assumed, would, in the absence of noise, have a uniform intensity throughout. The boundaries between the regions are assumed to approximately form discontinuities. Noise, either multiplicative or additive is assumed to corrupt these ideal images.

The algorithm is iterative with each iteration containing two rain stages. These being firstly the detection of probable edges and secondly the generation of closed region boundaries. These are now considered in more detail.

3.1 Detection of probable edges

The initial edge detection must overcome three major difficulties.

1. An automatic threshold must be selected which can reliably determine whether an edge should be set or not.

2. A sufficient number of pixels must be averaged to obtain an edge image which is relatively error free even for low contrast edges in the presence of high noise.

3. The spatial resolution of high contrast edges must be preserved.

These are dealt with as follows:

3.1.1 Selection of an automatic edge detection threshold.

The edge detection process is composed itself of two parts. The first is a convolution between the image and some edge enhancing mask. The output from this stage produces an image of edge strengths. A second stage is then applied which thresholds the edge strength image. Points with values above the threshold are set to 1 whilst those below are set to 0. This generates a second binary edge image. It is the selection of the threshold which is of interest here. Several threshold selection techniques are given in the literature. One of the commonest is to simply set the top 5% of the edge strength image to value 1 in the binary edge image. If we consider an image which is in reality a single region then clearly there should be no edges. However if the scheme outlined above is used edges would be generated. This is clearly undesirable.

Alternatively an absolute threshold might be set. Unfortunately such as scheme would not allow the algorithm to be generalised. What is required is some method of estimating when the variations within a given region in the image are due solely to the noise. This is achieved as follows.

A standard deviation is calculated for each region in the image which is given by

$$\sigma = \left(\sum_{i} \frac{(I_i - \hat{I}_n)^2}{N-1} \right)^{\frac{1}{2}} \qquad (3)$$

where $\hat{I}_n$ = averaged intensity for the region n and
N = number of pixels in region n.
(Initially the full image is taken as one region).

The output from the edge operator is then divided by the standard deviation of the region in which the centre of the edge operator mark lies. This yields an edge strength normalised to unity standard deviation. The normalised edge strength may then be compared to an absolute fixed threshold. The fixed thresholds are estimated initially by comparing the algorithms performance to human performance, but the final thresholds are fixed by the algorithm itself.

The human eye can detect small regions in an image provided these regions have a high region to background contrast. Larger regions may be detected for lower contrast ratios. In order to roughly match the algorithms performance to the human performance one particular region size is chosen. This is a 20 pixel by 20 pixel square. A synthetic image is then generated containing regions of this size with varying intensities on a background of unity intensity. Noise is then allowed to corrupt the image. The intensity ratio (R), at which the eye can just detect the regions, is determined.

As will be shown later computational considerations mean the algorithm is slightly sub-optimal. Consequently it may be expected that the algorithm will not work up to the standard generated by the eye. The intensity rating (R) above therefore, may need to be relaxed somewhat, thus generating a second ratio R. R will be fixed by the algorithm.

The algorithm (see later) works with several length scales. One of these scales (15) roughly matches the region sizes (20 x 20) considered above. This part of the algorithm is run on the test image. A relaxed ratio R' and an edge detection threshold are consequently chosen to satisfy the following. Of all the edges in the image 50% are found and simultaneously of all the points set 50% are set on the true edges. This fixes R' as 1.04R also sets the threshold for this length scale in the algorithm. It may be seen therefore that R is merely a guide to set R' rather than an absolute parameter in itself. However it is also apparent that the values of R and R' are similar. This suggests two possible conclusions. Firstly the algorithm should match the eye's performance closely. Secondly as the algorithm for computational reasons (see later), is set up in a sub-optimal way the eye may in fact not offer an absolutely accurate segmentation.

A new synthetic image is now generated consisting entirely of noise. The length scale considered above is run on the image and the number of edges (all necessarily false) noted. This turns out to be 0.22% of pixels in the image. The thresholds for the other length scales are set so as to generate the same percentage of false edges. The question of thresholds will be considered again in section 3.4.

The noise which as been allowed to corrupt the test images used above should ideally be representative of the noise found in real images. The following image types are considered to be of importance:

1. Coherent images where pixel values represent the intensity of the radiation incident on the receiver.

2. Coherent images whose pixel values represent the square root of the intensity of the radiation incident on the receiver. (The dynamic range of most imaging systems, including the eye, is too small to accomodate the type of imagery described above. However the problem may be overcome by displaying the square root of the intensity.)

3. Type 1 images displayed at a lower resolution with each low resolution pixel representing an incoherent average over four neighbouring high resolution pixels.

4. Type 2 images averaged as described in 3.

5. Incoherent imagery.

Coherent images are corrupted by multiplicative noise and additive noise and incoherent images by additive noise. However the additive noise component in coherent imagery is generally small. Indeed for the results given later the additive noise component is roughly three orders of magnitude smaller that the multiplicative component. The number of pixels falsely set as edges in a uniform test image will depend on the particular noise distribution given to the test image. These error rates are virtually identical for image types 4 and 5 and very similar for types 2, 3, 4 and 5. The results for type 1 images do show a change but the change is still small enough to allow the algorithm to produce satisfactory results. Having said this the output from the algorithm will of course be optimised if the correct image statistics are used.

The human eye has difficulty in segmenting down to the one pixel level in type 1 or type 2 images due to the high multiplicative noise (speckle). Consequently image types 3, 4 and 5 are the ones mainly considered.

3.1.2 The detection of low contrast edges without loss of resolution.

It was stated above that if a noisy image is viewed it soon becomes apparent that the eye can detect small regions only if there is a high contrast between the region brightness and the background brightness. As the region size increases the human observer will accept regions as being distinct for lower contrast ratios. This may be stated in a slightly different way. High contrast ratios enable a region boundary to be placed with a high spatial accuracy whilst low contrast ratios lead to a poorer spatial accuracy. This single fact leads to the main basis of the edge-detection portion of this algorithm.

When first presented with a noisy image the eye tends to be drawn to the bright areas. This suggests the algorithm should do the same. Consequently the techniques first detects very high contrast edges using edge enhancement operators with small length scales. Lower contrast edges are then detected with larger operator masks.

The interaction between the various masks is controlled. The reason for this will become apparent later.

Each edge enhancement operator is composed of a pair of windows of size N x M as shown in figure 2 (N and M are odd). The intensity difference across X - X is determined and compared to the threshold for this size of window. If the difference exceeds the threshold the pixel (i, j) is set to 1, otherwise to 0. The same procedure is carried out across Y - Y and the final value for (i, j) is calculated by applying an OR function to the two results.

3.1.2. i. The mask sizes and shapes.

It is computationally expensive to use masks at many orientations. Consequently only two orientations are used vertical and horizontal. This restriction may of course lead to difficulties when boundary orientations do not match mask orientations. However if the masks are made long and thin then these difficulties may be almost entirely removed. When $(N - 1)/2 \simeq 2M$ then for the worst case, a line at 45° to the horizontal, only ~5% of the mask area is rendered unusable. Unfortunately when a mask is long and thin its length can make it a little unwieldy in a complicated image. Ideally then more angles should be used thus allowing the above condition to be relaxed to $N \simeq M$. Such as window would have a compact shape. This extension however is not used here for reasons of computational speed.

The next question to be answered is what should the relative sizes of the various windows be. The windows average over $\left(\frac{N-1}{2}\right) \times M$ pixels. Consequently this quantity should be considered when comparing the size of different windows. To obtain an answer to the above question we again appeal to the results obtained by eye. By considering various background to region intensity differences for a given region size it is found that the eye may only determine the value of a significant difference to within approximately 20%.

As region size and intensity ratio are related this also implies an error in determining a significant (ie detectable) region size of ~ 20%.

Consider window sizes N x M and $\sqrt{2}$N x $\sqrt{2}$M applied to a region of width N $\frac{(1+\sqrt{2})}{2}$ As the region size matches neither window perfectly then some information is lost. The amount lost amounts to approximately 15-20% of the total window area. Consequently the window averaging areas are set to have ratios of ~2.

The possible set of windows to be used is given in table 1. There are two points here worthy of further comment. Firstly it will be seen that the ratio $\left(\frac{N-1}{2}\right)$:M is significantly less than 2 for the 3 x 3 and 7 x 3 windows. It would therefore be beneficial to change the 3 x 3 windows to a 3 x 3 Sobel operator. For the 7 x 3 window, however, only 11% of the available window area is lost for a boundary at 45° to the horizontal. The benefit which might be obtained by implementing a 5 x 5 Sobel operator instead is therefore not so obvious. Considering the way in which windows of varying sizes interact (see section 3.1.2.ii) the use of a 5 x 5 Sobel detector would probably be detrimental to the algorithms performance. The 7 x 3 window is therefore retained as shown, but the 3 x 3 window is replaced by a 3 x 3 Sobel edge operator.

The second point of interest in table 1 is that two windows have equal values of $\left(\frac{N-1}{2}\right)$ x M. This occurs for two reasons. As M is always odd it is difficult to satisfy the condition for a doubling in $\left(\frac{N-1}{2}\right)$x M whilst still maintaining $\left(\frac{N-1}{2}\right) \simeq$ 2M. Additionally the value of $\left(\frac{N-1}{2}\right)$ x M for this window is approximately the central value in the geometric series of window sizes. Consequently this window is of particular importance as it is likely to match many length scales in an image. The use of two window shapes for this size is an attempt to overcome the somewhat unwieldy behaviour of the windows for this important length scale. Having said this it may still be the case that the 31 x 3 window is superfluous. This has yet to be determined.

3.1.2. ii. The interaction between different masks.

It has been stated so far that the masks described above are used in an hierarchial manner. The particular form of this implementation will now be considered in more detail.

Initially the 3 x 3 Sobel operators are applied to the image thus generating an edge image consisting of 0's and 1's. Wherever the density of 1's is low the 7 x 3 masks are then applied. More edges (1's) are generated and combined with the existing ones via an OR function. The subsequent masks are then applied in sequence.

It will be noted that masks are only applied where smaller masks have been mostly unsuccessful in detecting edges. This is essential. If the requirement was removed then bright areas would generate false alarms in the larger masks due to the decrease in the thresholds used for these masks. Specifically this protection is performed as follows.

If edge pixels are encountered in a mask then only part of the mask is used (see figure 3). Whenever a region has enough contrast for its boundary to be detected by a smaller window then edge pixels will be set. It may be seen therefore that the use of the protection indicated in figure 3 will stop the edge enhancement operator overlapping the already segmented regions. However when a gap in a boundary is encountered the operator will still function.

Clearly as we begin to reduce the number of pixels contributing to the edge operator the error rate will increase. Therefore some limit must be placed on the allowable reductions in size. It will be recalled that the eye cannot determine a minimum detectable region size to better than approximately 20%. If the number of pixels on each side of an operator is allowed to drop by 20% then the error rate for falsely setting edges on a blank image rises to 0.64%. The error rate for the full window being 0.22%. The new error rate is still small enough for the algorithm to cope with. Consequently neither the new error rate nor the effective change in threshold parameters (generated by a change in operator size) represent significant changes.

False edges in an image occur whenever the noise processes conspire to generate a large value in the edge strength image. Very large deviations of this sort are detected by the smaller windows in the algorithm. Subsequent edge operators are therefore protected from the effects of large noise induced deviations from the image mean. It is assumed that the noise in the image is independent of position. Consequently we are unlikely to encounter two areas, with large noise values close together. Hence it is reasonable to allow the edge operators to be heavily reduced in size whenever edges, created by previous operators, are encountered.

The program is now run in full on a test image of just noise. If a 75% reduction in operator size is allowed then the number of pixels set for each pair of windows (false edges) is found to represent ~ 0.3 - 0.5% of the total number of pixels in the image. This is an acceptable error rate for a single pair of edge operators according to the discussion above. The total number of false edges (from all the edge operators) is found to be ~ 4% of the image. It has been determined that this density of false edges is also acceptable to the algorithm.

A reduction in edge operator mask sizes of 75% is appealing in other ways as well. As the areas of consecutive edge operators have a ratio of 1:2 then the reduction of 75% described above allows any pixel to be tested by at least two edge operators even in a complicated image. In addition to this advantage the possible shapes an operator mask may take would appear to be beneficial. In particular pixels may be protected within a mask such that the effective overall length of the window (Neff) and the effective overall width (Meff) are equal. In the absence of orientation problems such a window shape would be most useful due to its compactness. Fortunately the method by which pixels are removed from the active mask area does greatly reduce the orientation problem.

As a result of all of the arguments given above an area reduction due to the existence of previously set edges, of 75% is allowed in the edge operator masks.

3.1.3 The detection of small regions.

Many images contain information down to the one or two pixel level. However the edge operators described above have difficulty in dealing with such regions. Such regions are set as all edge (see figure 4). Thus all information about the small region is lost as it now looks like one broad edge. To overcome this problem a set of operators have been designed to detect such regions explicitly. The operators are as shown in figure 5. The following algorithm is used on the outputs

```
IF (($A \neq B$).AND.($B \neq D$))THEN
    SET $B_{(n+1)/2}$
IF ($C \neq D$) THEN
    SET $C_{(n+1)/2}$
```

The operation $\neq$ is taken here as being identical to the operation in the larger edge operators for determining whether an edge was present or not. That is, depending on the value of n (A_n etc), a threshold is set at some intensity difference. Edges are deemed to exist when the threshold is exceeded.

The pixels set are then taken as regions in their own right rather than edge pixels. This algorithm detects most of the one and two pixel wide regions. It also ensures that the edges of larger regions are not falsely set as being separate one pixel wide regions. Linear features up to 7 pixels in length are also detected to some extent but not particularly efficiently.

3.2 The generation of closed region boundaries

In a noisy image any edge detection technique will generate a broken edge image. The second stage of the algorithm generates an interpolation between the broken edges thus producing closed regions.

Consider the edge image shown in figure 6. There are 5 regions in the image as indicated. The algorithm should be designed so as to find these regions. It works as follows, initially disc like templates (see figure 7) are stepped over the image. The number of edge pixels included within the template boundary is monitored. Whenever this number is zero the average value of the original image covered by the template is written to an average image file. The final output of this stage of the algorithm is therefore an average image which contains the mean pixel values of the original image where no edges were found. The average image is unset elsewhere. Such an output is indicated in figure 8.

A series of average images is obtained for various template sizes. Eight disc like templates are used with M x N values of 64 x 8, 48 x 8, 32 x 4, 24 x 4, 16 x 12, 12 x 2, 8 x 1, 6 x 1. These are stepped over the image in 8, 8, 4, 4, 2, 2, 1, 1 pixel steps respectively. The disc templates are followed by 4 square templates of size 4 x 4, 3 x 3, 2 x 2 and 1 x 1. These are stepped using 1 pixel increments. The particular sizes of the templates will be considered later. We now have 12 average images. These are combined in the next stage of the algorithm.

The output from the largest template averaging operation is now taken. Pixels which have been set and are adjacent are now grouped together thus forming the image shown in figure 9. There are now 3 regions. These regions may now only be grown. That is to say none of these 3 original regions may be joined together nor may they be split at any subsequent stage of this boundary generation portion of the algorithm. It should be noticed that no account is taken of the average values of the set pixels (shaded in figure 8) at this stage. Adjacency of set pixels is the only criteria for joining. It will be recalled that edge pixels prevent joining across a region boundary.

The average output derived from the next template is now added to the current groupings shown in figure 9. Pixels which have been set in this second averaged image and are adjacent to pixels in one of the 3 regions shown in figure 9 are grouped together with the relevant original regions. Pixels which may not be grouped thus are allowed to start new separate regions. Pixels touching more than one of the original 3 regions are joined to that region for which the average values are the closest match. The output from this stage is shown in figure 10. The process is continued until all the average images have been used. Finally the edge pixels are combined into the regions in a similar manner.

It will be recalled from section 3.1.3 that small regions have been detected directly. These are combined into the overall image in a virtually identical manner to that described above. The main differences being as follows. Firstly the average images are set onl y where the averaging templates entirely overlie edge pixels set by the small region detection process. Secondly due to the size of the regions only 3 x 3, 2 x 2 and 1 x 1 square averaging templates are used.

As a result of the processes outlined above the segmented image should now consist of a set of closed regions whose boundaries hopefully approximate the real boundaries in the image fairly accurately. However due to the approximations taken in forming the disc shapes and the sizes of the steps used to scan the image we might expect slight errors to be made. It has been found that regions may become broken due to the particular placing of one or two noisy edge pixels. Consequently each boundary is tested for the number of edge pixels it contains. If the boundary is not supported over at least 20% of its length by edge pixels it is removed.

3.2.1 The relative sizes of the averaging templates.

The different templates are used to stop leakage between regions. If the contrast across a boundary is low then the boundary will be heavily broken. However for the boundary to be significant it must separate two large regions. Consequently if a large essentially edge free area can be found a large gap in any edge must occur before we allow leakage away into another region.

Consider figure 11 which depicts the action of the 2 types of disc operators on a broken edge. It may be seen for figures 11ai and 11aii that a "boundary" may be penetrated only if edge pixels occupy less than approximately 67% and 50% of the boundary respectively. The percentages of edge pixels required to avoid breaking the boundary in figures 11bi and 11bii are 50% and 33%. These figures are chosen as to roughly match the decision point at which a human interpreter may consider a set of edge pixels to constitute a boundary (50%). The ratios of successive template sizes are chosen so that boundary breaking on the new template begins to occur roughly where is ceased for the previous template as depicted in figure 12. This fixes the relative sizes of the templates.

3.3 Iteration of the algorithm

In order to iterate the algorithm some measure should be devised to determine whether the pixel intensity variations within a region in an image are due entirely to noise. This measure is taken to be the standard deviation averaged over the image as described below.

Initially is is uncertain whether the standard deviation measured for a given image is due to its noise characteristics alone or whether variations in underlying intensity offer significant contributions. The standard deviation measure should therefore be able to distinguish between these two situations. Consider an image region (I_k) corrupted by multiplicative noise (N)

$$I_k'(i,j) = I_k(i,j).N(i,j) \qquad (4)$$

The underlying intensity is assumed to be constant for any single region. This is the world model adopted for this edge detection algorithm. Hence

$$I_k'(i,j) = I_{ko}.N(i,j) \qquad (5)$$

The noise is assumed to have a mean of 1 and so the normalised standard deviation is found to be

$$\sigma_k = \left(\sum_i \frac{(N(i,j)-1)^2}{n_k}\right)^{\frac{1}{2}} \qquad (6)$$

where n_k is the number of pixels in the region k. An overall measure of the standard deviation may be obtained by taking the mean value of the normalised standard deviation for each image point (i,j). This is (for an image with m regions)

$$\sigma_T = \sum_{k=1}^{k=m} \frac{n_k \sigma_k}{\sum_{k=1}^{m} n_k} \qquad (7)$$

The value of this measure should reach a minimum when the image is correctly divided into its component regions. Consequently σ_T is taken as a valid measure of the algorithms progress. After each pass of the the algorithm σ_T is calculated. If the new value for σ_T is less than the previous value the program is iterated again.

The algorithm necessarily runs at least once. This is because no prior knowledge of the expected image statistics is required. Such a generalisation allows the program to be run on images where the statistics are not known or are known only approximately.

3.4 Post-processing

Several approximations have been made in the algorithm to allow for faster computation. Among the approximations were the use of only two edge operator orientations, the use of only a finite set of edge operators and the use of only a small set of area averaging templates (section 3.2 and 3.2.1). Consequently we might expect the algorithm to behave slightly suboptimally. Specifically we might expect it to generate some false edges. Post-processing is therefore used to remove these. In order to decide when we should join two regions together we appeal to the results of the edge detection thresholds measured in section 3.1.1.

The threshold (T) applied to the edge strength image for an edge operator with a half mask area of H is found to be proportional to $\sqrt{H}$:

$$T = K \sqrt{H} \tag{8}$$

The edge strength image (S) itself is given by

$$S = H \mid (\mu_{NL} - \mu_{NR}) \mid / \sigma \tag{9}$$

where μ_{NL} and μ_{NR} are the mean values of the pixels contained in the two halves of the edge operator and σ is the standard deviation of the region in which the operator is centred.

Edges are set when S > T or

$$\frac{|(\mu_{NL}-\mu_{NR})|}{(\sigma/\sqrt{H})} > K \tag{10}$$

K is found to be 4.6.

Consider now two adjacent regions in the final image. We take the case which matches the calculation above. The two adjacent regions are therefore assumed to have equal sizes H and have similar means and standard deviation. The following condition must be satisfied if the regions are to be left unjoined.

$$|\mu_1-\mu_2| > \alpha \frac{\sigma_1+\sigma_2}{\sqrt{H}} \qquad (11)$$

where μ_i and σ_i are the mean and standard deviation for the regions. We assume $\sigma_1 + \sigma_2 \simeq 2\sigma_1$ and hence

$$\frac{|\mu_1-\mu_2|}{(\sigma_1/\sqrt{H})} > 2\alpha \qquad (12)$$

Therefore the two conditions (10 and 12) are the same if $\alpha = K/2 = 2.3$.

In testing the algorithm on various images whose statistics are approximately known it is found the final overall measure of the standard deviation is ~ 40% above that expected for pure noise. It is assumed therefore that each region has a standard deviation which is too large by ~40%. Consequently α is taken as $(K/2)/1.40$ to compensate.

RESULTS

The algorithm is designed to be a general purpose low-level segmenter for any type of two-dimensional image. However it will be recalled from section 3.1.2 that the output may be optimised if the algorithm is trained on synthetic data having the same noise statistics as the images of interest. The algorithm has been specifically trained on type 4 images (section 3.1.2). Results obtained on such images might be expected therefore to be better than results obtained from other image types.

The first sets of results are shown in figure 13. The original images (13ai, 13bi, 13ci) are all examples of type 4 imagery. The three images are chosen so as to represent as wide a range of length scales and image scenes as possible to the algorithm. The images in 13aii, 13bii, 13cii represent the output of the algorithm where each region is displayed with a value equal to the average value of the pixels held in that region. Figures 13aiii, 13biii, 13ciii show the final edge maps for the same regions.

Each original image contains 256 x 256 pixels (65536 pixels) whereas the final outputs contain ~1000 to ~2000 regions. Thus a large data reduction has been achieved without the loss of much structural information. This level of data reduction whilst still retaining much of the information is probably sufficient to allow higher level pattern (shape, texture, etc) recognition processes to be used.

In addition to the results obtained using type 4 images, images from the other image classes have been segmented. The algorithm, and the parameters used in the algorithm are identical to those used to generate the results in figure 13.

Figure 14 shows the output from the segmentation of an image of type 5 (incoherent data). It will be noticed that many regions have been generated. This is due to the fact that the additive noise is small. Consequently the algorithm is capable of detecting subtle changes within the image. The low noise also allows the algorithm to detect small regions fairly easily. However due to the problems outlined in section 3.1.3 many such regions become broken or lost. The problem may be overcome by resampling the image (expanding it). The subsequent segmentation (figure 15) is then found to avoid the problems associated with small regions.

The algorithm is capable of generating a great deal of information when the noise content of an image is low. This is the case here with the incoherent data. It is of course possible to reduce the number of regions and information by raising the edge detection thresholds. This may be beneficial in some cases. However it is probably better in general to allow such data reduction to be performed by a subsequent higher level operation.

The algorithms performance on the remaining 3 classes of image is shown in figures 16, 17, 18. Even though the algorithm has not been optimised to run on such images the results are still acceptable. If prior knowledge is available about the image statistics then the results may be slightly improved. The improvement for type 1 images is expected to be the greatest and may in fact be significant. It is interesting to note that the eye also has great difficulty in segmenting this type of image. Consequently we might expect the algorithm to reflect such behaviour.

POSSIBLE IMPROVEMENTS TO THE ALGORITHM

Three possible extensions to the algorithm are proposed:

1. It is possible to implement the algorithm in parallel. The maximum degree of parallelisation possible is roughly equal to the number of pixels in the image. With the advent of the transputer a large degree of parallelisation would indeed appear feasible.

2. The algorithm iterates by calculating a total measure of the standard deviation. This method of measuring algorithm performance has its obvious disadvantages. The most significant of these is that improvements in the boundary accuracy of large regions tend to mask changes in the smaller regions. To rectify the situation each region should be tested separately. Preliminary results using this measure of segmentation accuracy suggest significant improvements might be made.

3. Errors in the edge image (incorrectly set edge pixels) are at present left untouched. The template averaging is then invoked to deal with them. Preliminary results, however, suggest it may be possible to remove many errors before this stage by the demanding that each edge pixel forms part of an extended edge. The improvements obtained, according to the preliminary results, again appear to be significant.

6. CONCLUSIONS

An algorithm has been presented which generates good low-level segmentations of various types of image data. The results obtained are qualitatively better than those obtained using previously reported segmentation algorithms apart from the Bayes reconstruction methods [ref 6]. The outputs from the Bayes Recontruction and from this algorithm appear similar. However Bayes reconstruction is computationally expensive.

Possible extensions to the algorithm would appear to offer potentially significant improvements in the segmentation.

The output from the algorithm appears to preserve a very high proportion of the structural information in the image. This, coupled with a data reduction of approximately 98% should enable higher level pattern recognition techniques to be used. Such an approach has previously been impossible for high noise imagery.

REFERENCES

1. Wen-Hsiang Tsai, Shiaw-Shian Yu.
"Attributed String Matching with Merging for Shape Recognition."
IEEE Transactions on Pattern Analysis and Machine Intelligence.
Vol PAMI-7 No 4, July 1985, pp 453-463.

2. V S Frost, K S Shanmugan, J C Holtzman.
"Edge Detection for Synthetic Aperture Radar and Other Noisy Images."
International Geoscience and Remote Sensing Symposium (IGARSS) 1982,
WA-5, 2, pp 4.1 - 4.9.

3. J J Gerbrands, E Backer.
"Split-and-Merge Segmentation of SLAR-Imagery: Segmentation Consistency."
IEEE 7th Int Conf on Pattern Recognition, Vol 1.
Montreal, Canada, July 30 - Aug 2, 1984 (pp 284-286).

4. Ting-Chuen Pong, Linda G Shapiro, Layne T Watson, Robert M Haralick.
"Experiments in Segmentation Using a Facet Model Region Grower."
Computer Vision, Graphics and Image Processing, 25, 1-23 (1984).

5. W E L Grimson, T Pavlidis.
"Discontinuity Detection for Visual Surface Reconstruction."
Computer Vision, Graphics and Image Processing, 30, 316-330, (1985).

6. H Derin, H Elliot, R Cristi, D Geman.
"Bayes Smoothing Algorithms for Segmentation of Binary Images Modelled by Markov Random Fields."
IEEE Transactions on Pattern Analysis and Machine Intelligence.
Vol PAMI-6 No 6, Nov 1984, pp 707-720.

7. C J Oddy, A J Rye.
"Segmentation of SAR Images Using a Local Similarity Rule."
Pattern Recognition Letters 1 (1983) pp 443-449, July 1983.

8. Hon-Son Don, King-Sun Fu.
"A Syntactic Method for Image Segmentation and Object Recognition."
Pattern Recognition, Vol 18, No 1, pp 73-87, 1985.

9. A Rosenfeld, M Thurston.
"Edge and Curve Detection for Visual Scene Analysis."
IEEE Transactions on Computers, Vol C-20, No 5, May 1971, pp 562-569.

10. A Rosenfeld, M Thurston, Y-H Lee.
"Edge and Curve Detection: Further Experiments."
IEEE Transactions on Computers, Vol C-21, No 7, July 1972, pp 677-715.

11. D Marr, E Hildreth.
"Theory of edge Detection."
Proc R Soc Lond B 207, (1980), pp 187-217.

12. S C Giess.
"Edge Detection in SAR Imagery Using Gradient Operators."
RSRE Memorandum Number 3743 (1984).

13. B J Freitag, B Guindon, A J Sieber, D Goodenough.
"Enhancement of High Resolution SAR Imagery by Adaptive Filtering."
International Geoscience and Remote Sensing Symposium (IGARSS '83) San Francisco C A USA 31 Aug - 2 Sept 1983, pp WP 4.5/1-6, Vol 1 (Sept 1983).

14. A Hendry, S Quegan.
"Automatic Segmentation Techniques for SAR Images."
MSDS Laboratory, Marconi Research Centre, GEC Research, Chelmsford.

TABLE 1 Window sizes used for the edge enhancement operators

N x M	$\left(\frac{N-1}{2}\right)/M$	$\left(\frac{N-1}{2}\right) \times M$	Ratio to previous $\left(\frac{N-1}{2}\right) \times M$
3 x 3*	0.33	3	-
7 x 3	1.00	9	3.00
15 x 3	2.33	21	2.33
31 x 3	5.00	45	2.14
19 x 5	1.80	45	1.00
27 x 7	1.86	91	2.02
45 x 9	2.44	198	2.18
57 x 13	2.00	364	1.84

* REPLACED BY SOBEL OPERATOR (SEE TEXT)

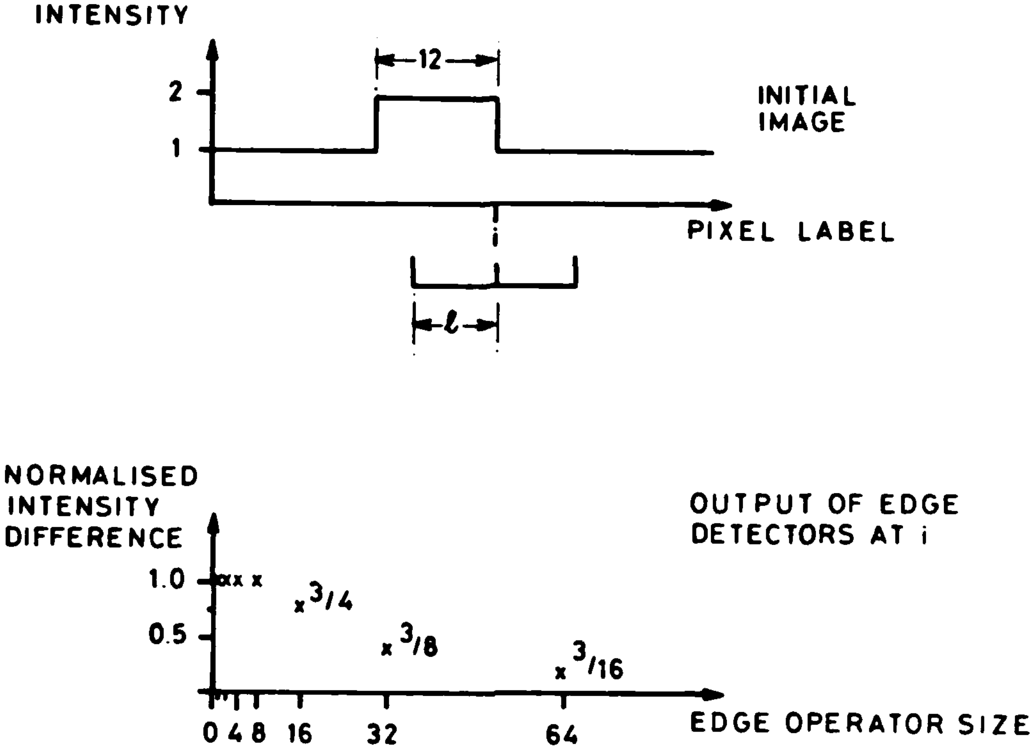

Figure 1. Optimum neighbourhood size selection for the edge detection scheme given by Rosenfeld et al [ref 10]. In the absence of noise a neighbourhood size of 8 would be selected.

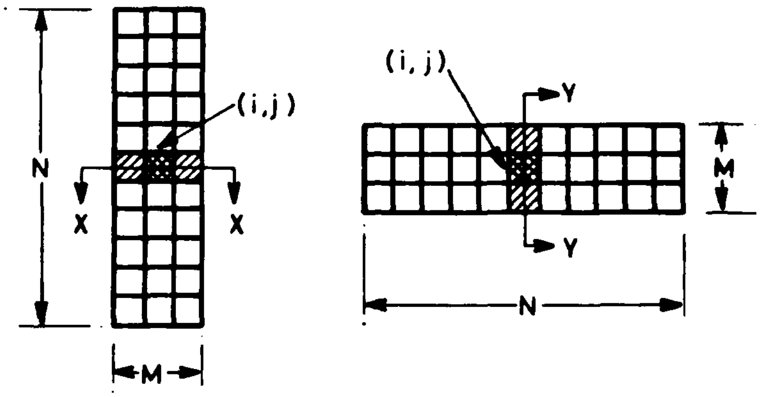

Figure 2. The segmentation scheme described in this paper begins by detecting possible edges. These edges are detected using edge enhancement operators as shown here. Shaded pixels are not used in the edge detection process.

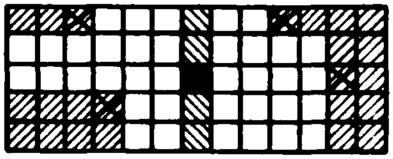

Figure 3. Whenever previously set edge pixels (marked by a cross) are encountered by an edge enhancement operator parts of the edge mask are removed from the calculation of an edge strength. These removed or protected pixels are shaded.

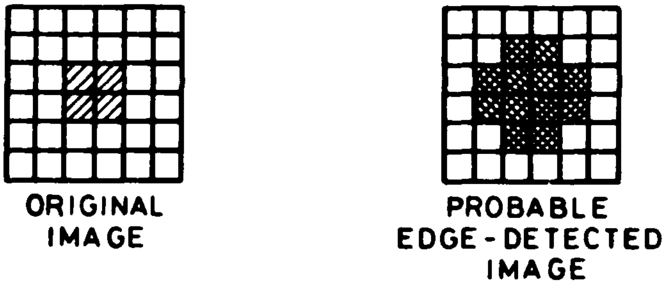

Figure 4. Edge enhancement operators acting on small regions generate an output which appears as all edge as indicated. This does not allow the small region to be detected by later processing techniques and the region is subsequently lost.

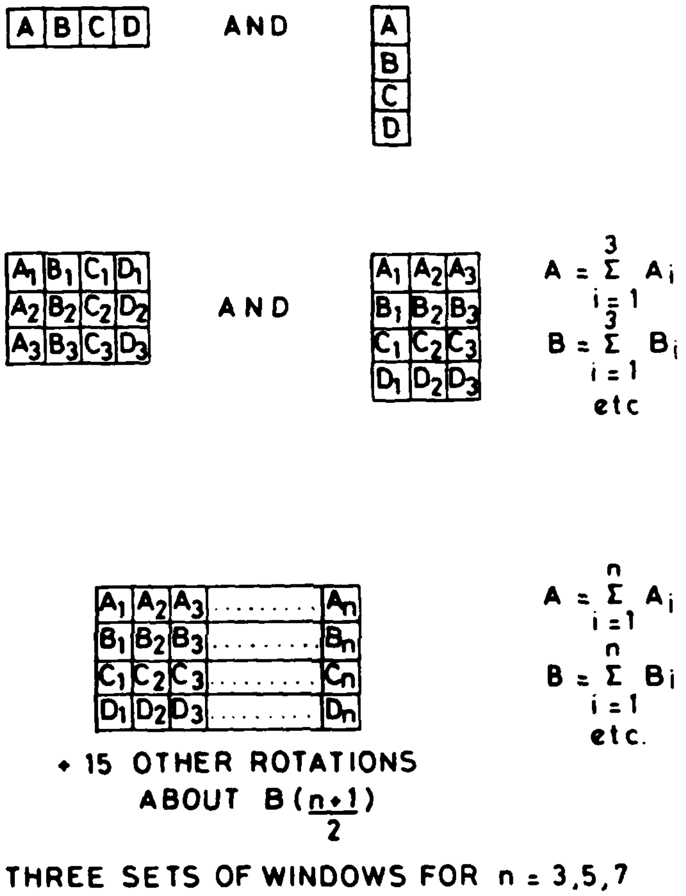

Figure 5. Small regions are detected directly by the use of the operators shown here.

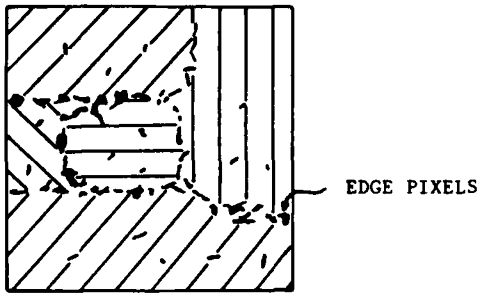

Figure 6. A typical edge image after the detection of possible edge pixels. Note both the false edges set in homogeneous regions and the broken boundaries.

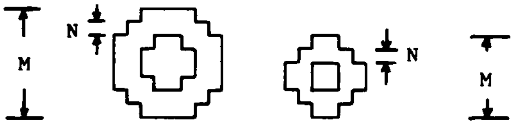

Figure 7. Possible locations of homogeneous regions are detected by using the disc-like templates shown in this figure. Regions are grown from the positions where such templates may be placed without touching edge pixels.

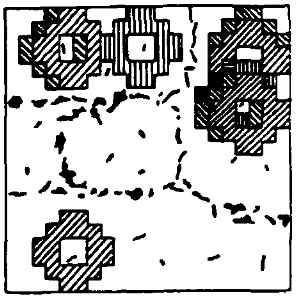

Figure 8. A typical average image formed by averaging over the areas fixed by valid positionings of the disc-like templates shown in figure 7.

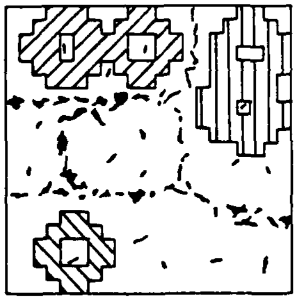

Figure 9. Regions are grown initially by using the average image (figure 8) for the largest disc template size (figure 7). Adjacent set pixels in the average image have been grouped together. Three regions have thus been formed.

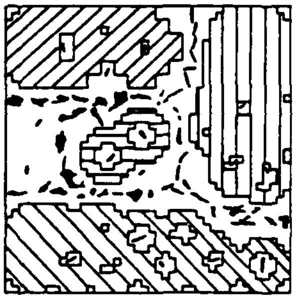

Figure 10. The initial regions shown in figure 9 are now added to by using the results from other average images. The region map shown here is a result of adding the next average image.

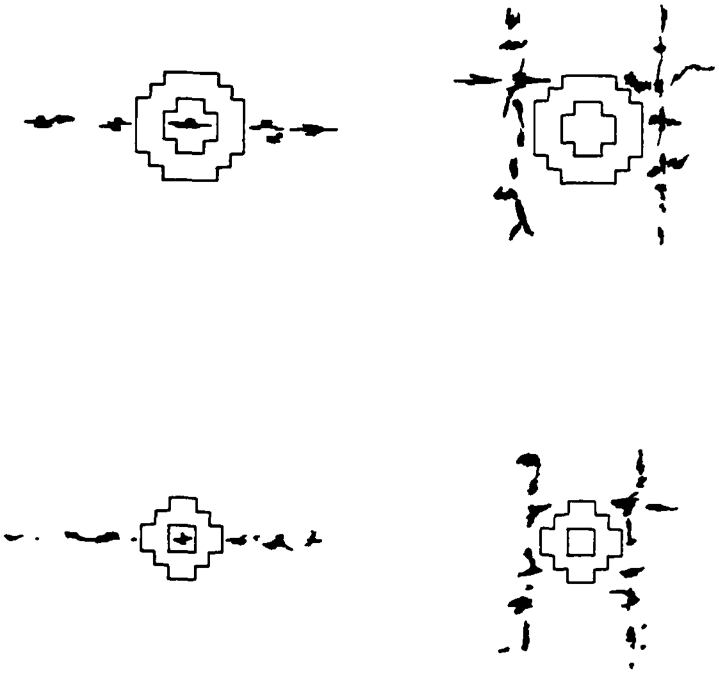

Figure 11. The disc templates shown in Figure 7 are used to create continuous region boundaries. Their action when dealing with a possible broken edge is shown here.

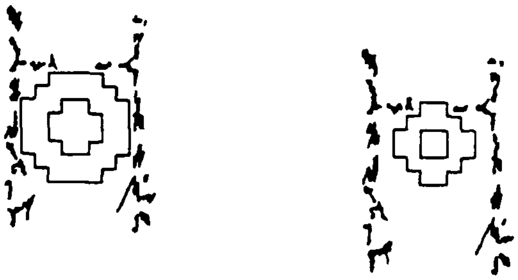

Figure 12. The amount of boundary leakage prevented by templates of differing sizes is shown here. It may be seen that as the larger template begins to stop breaching the potential edge at the point where the smaller template starts.

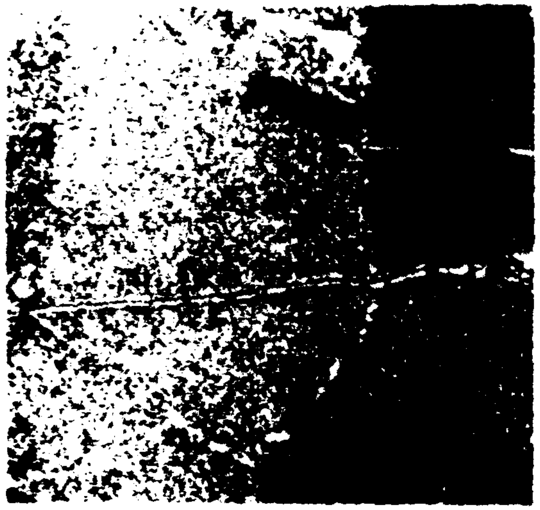

Fig. 13ai

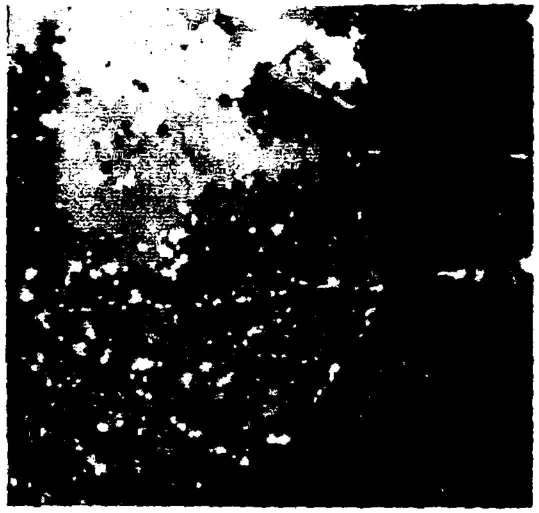

Fig. 13aii

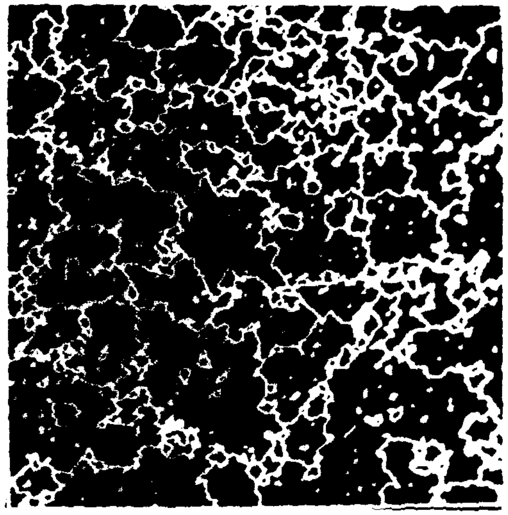

Fig. 13aiii

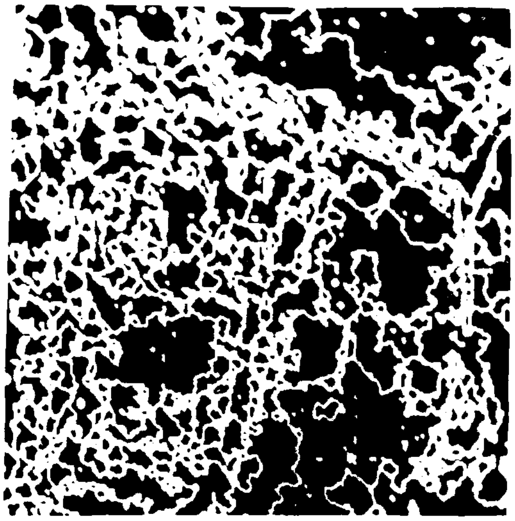

Fig. 13biii

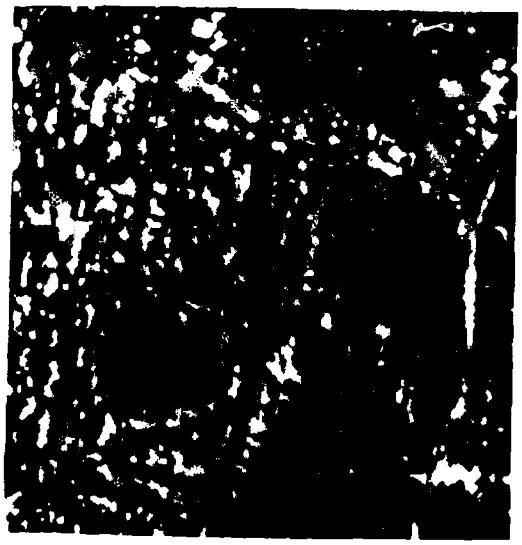

Fig. 13bii

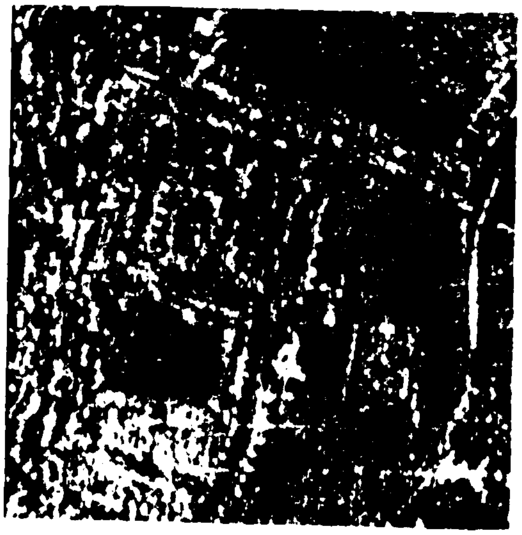

Fig. 13bi

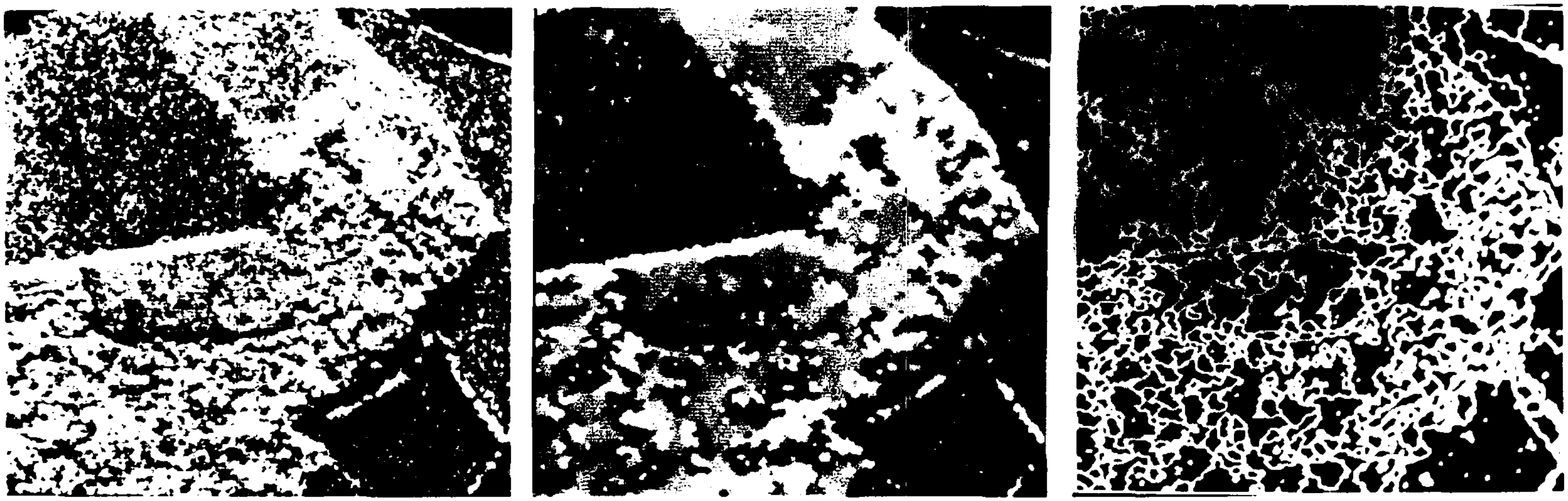

Fig. 13ci

Fig. 13cii

Fig. 13ciii

Fig. 13. The algorithm applied to type 4 imagery (see text). Three different images (a, b, c) are shown depicting a wide range of images subject and length scale. The images (i) show the original data whilst (ii) is the output of the algorithm with each region displayed with its average pixel intensity. The images (iii) show the region edge maps.

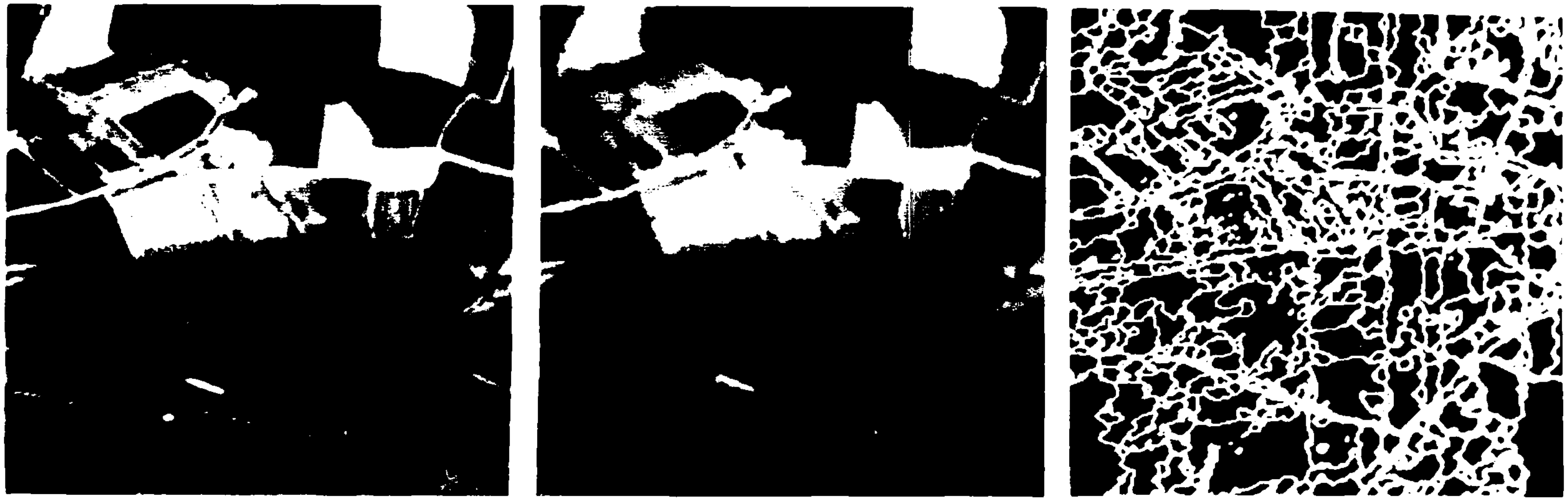

Fig. 14i

Fig. 14ii

Fig. 14iii

Fig. 14. The algorithm applied to an incoherent image (type 5 – see text). The various images correspond to those in Figure 13.

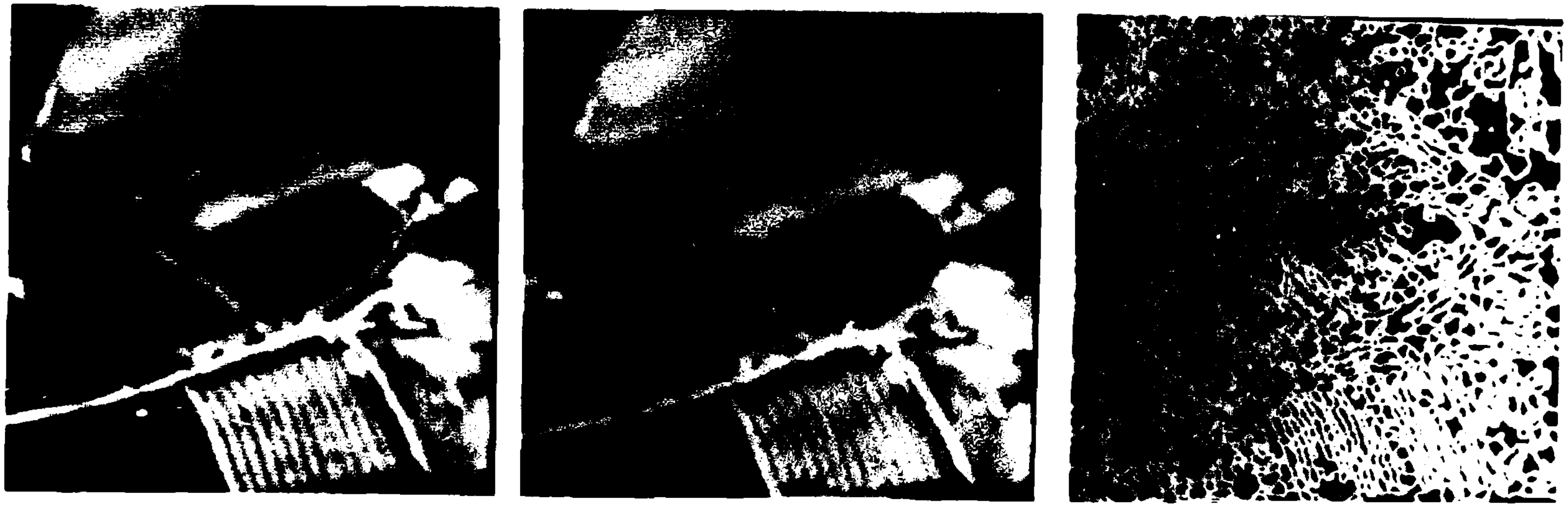

Fig. 15i

Fig. 15ii

Fig. 15iii

Fig. 15. Regions with length scales of the order of one pixel cause some problems for the algorithm. Part of the image in Figure 14 has therefore been expanded here and then segmented. Images are as before.

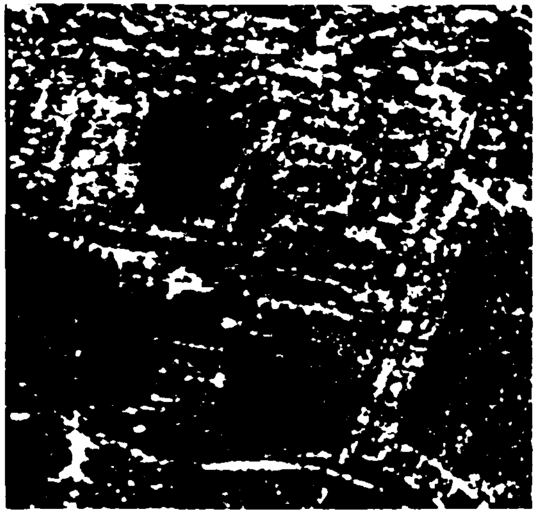

Fig. 16ai

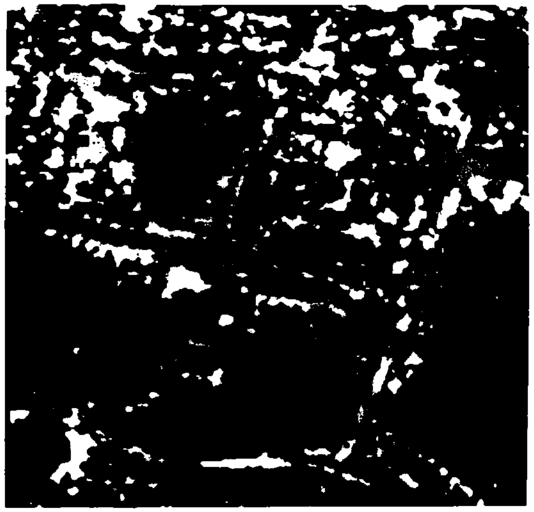

Fig. 16aii

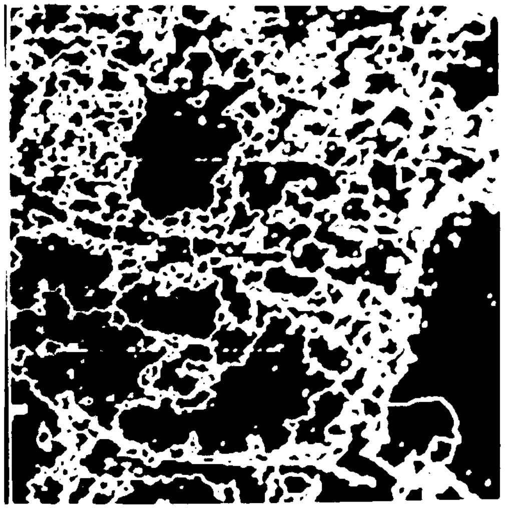

Fig. 16aiii

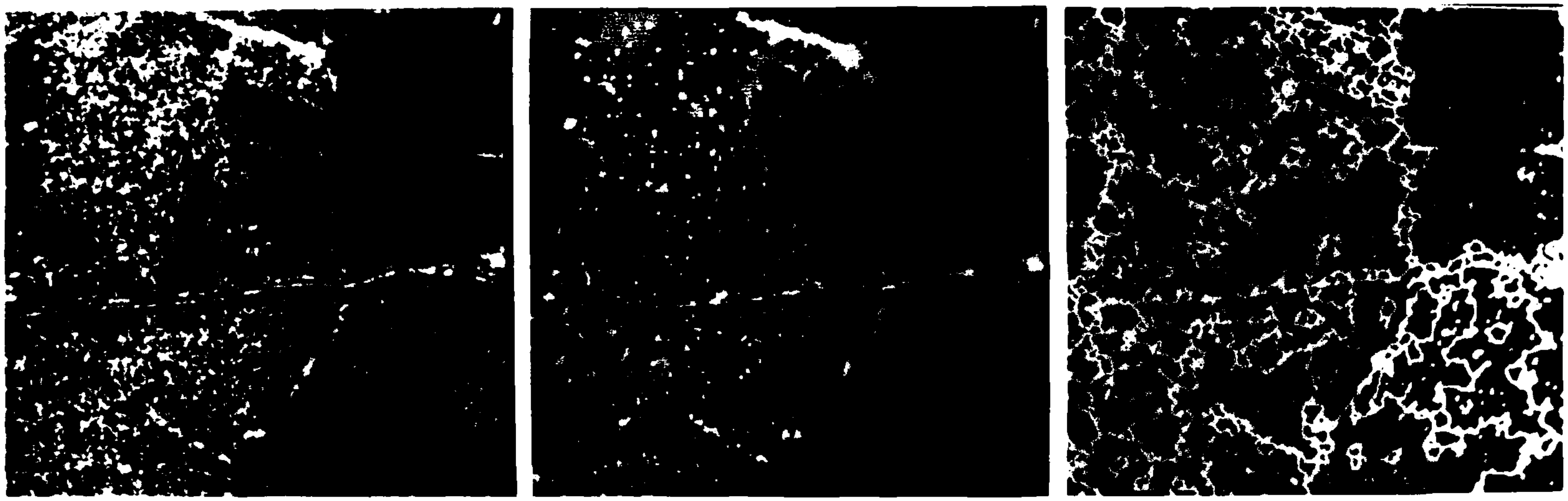

Fig. 16bi

Fig. 16bii

Fig. 16biii

Fig. 16. Results obtained using type 3 imagery (see text).

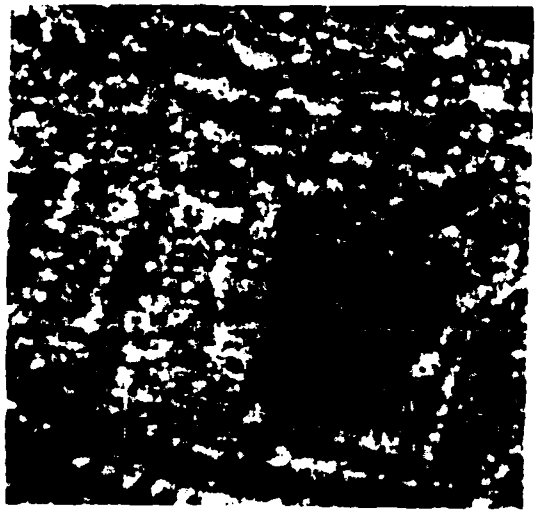

Fig. 17ai

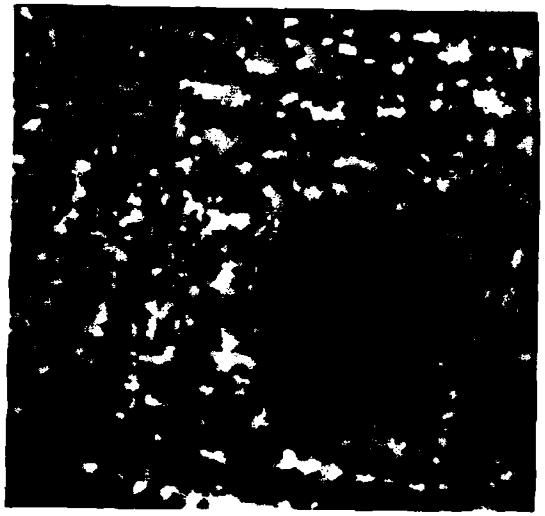

Fig. 17aii

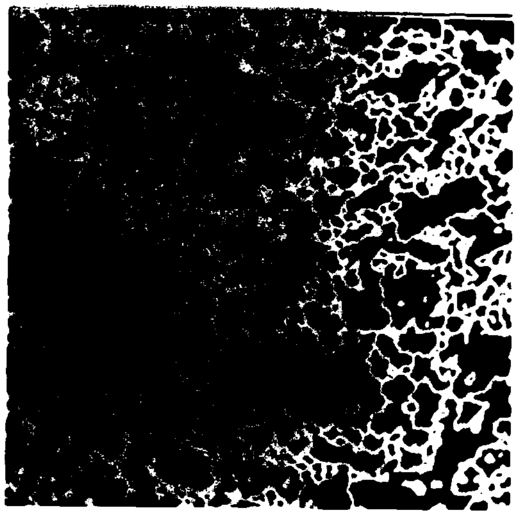

Fig. 17aiii

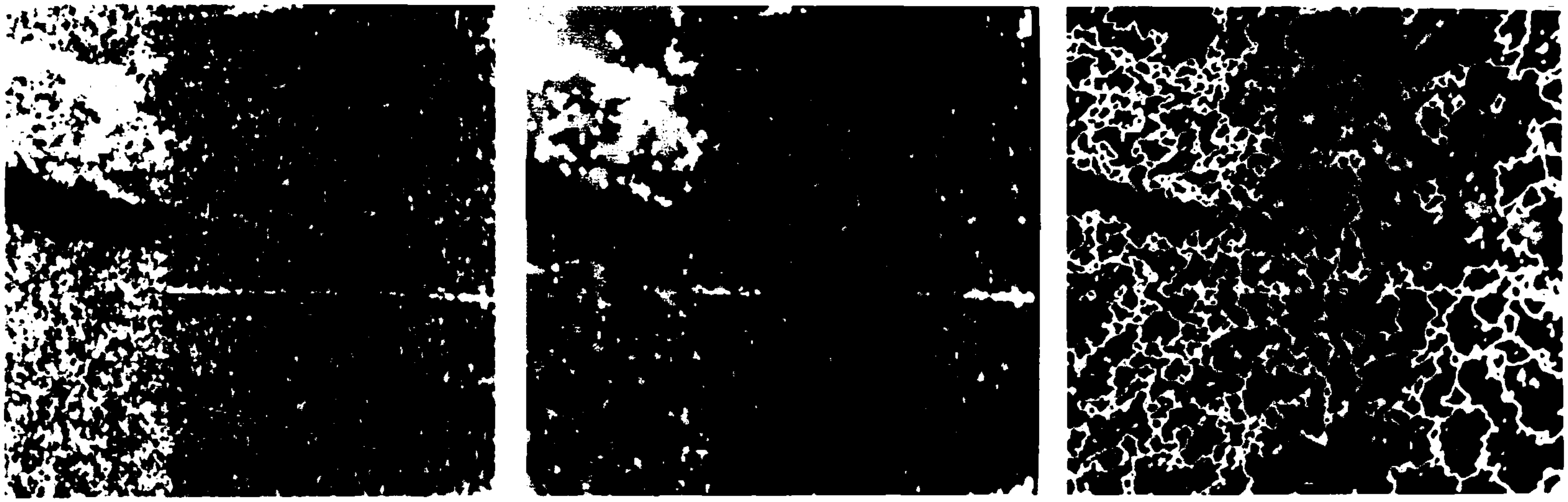

Fig. 17bi

Fig. 17bii

Fig 17biii

Fig. 17. Results obtained using type 2 imagery (see text).

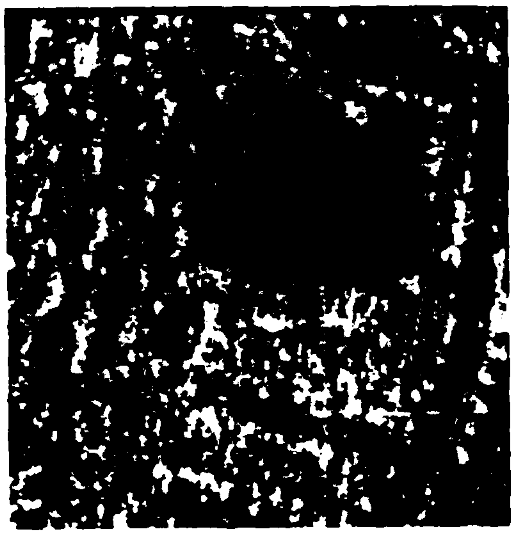

Fig. 18ai

Fig. 18aii

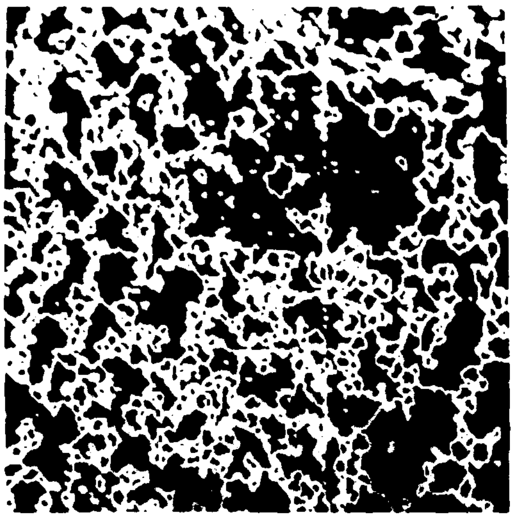

Fig. 18aiii

Fig 18bi

Fig 18bii

Fig 18biii

Fig 18 Results obtained using type 1 imagery (see text)

DOCUMENT CONTROL SHEET

Overall security classification of sheet .......UNCLASSIFIED.. ....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference Memorandum 3900	3. Agency Reference	4. Report Security Classification U/C

5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location Royal Signals and Radar Establishment
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title

Low level segmentation of noisy imagery

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials White R G	9(a) Author 2	9(b) Authors 3,4...	10. Date	pp. ref.

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement

UNLIMITED

Descriptors (or keywords)

continue on separate piece of paper

Abstract A possible approach to image segmentation is first to perform a 'low-level' segmentation. This then allows an original image to be described in terms of a set of simple regions or primitives. Objects in the image may be subsequently recognised by matching these primitives to patterns of primitives in a data-base. It is found that current techniques for low-level image segmentation fail when applied to high noise images. An algorithm is presented which overcomes the problems associated with high noise and succeeds in generating low-level segmentations of noisy imagery. The algorithm is shown also to work on low noise data.